intel®

July 1986

82588 Design Kit Users Guide

Order Number: 296098-001

Additional copies of this manual or other Intel literature may be obtained from:

Intel Corporation
Literature Distribution
Mail Stop SC6-714
3065 Bowers Avenue
Santa Clara, CA 95051

82588 DESIGN KIT USERS GUIDE

CONTENTS

1 INTRODUCTION

The purpose of this Design Kit is to present the user with a "Hands on Experience" introduction to the 82588 and to StarLAN. The Kit Board (from now on referred as StarLAN board) is meant to be a simple and powerful vehicle for evaluation of the 82588 and the emerging StarLAN standard (802.3 1BASE5).

This guide will provide insight into the assembly, operation and functioning of the kit as well as explain the basic circuits implementation. Also, detailed assembly instructions will be provided together with troubleshooting guidelines in case of trouble.

Before we continue with the kit explanation please check that your package is complete and contains the following material:

ATTENTION

The 82588 included in your kit is a metal-oxide-semiconductor (MOS) integrated circuit and can be permanently damaged by static electricity. Do not remove this component from its protective, black foam backing until instructed to do so.

COMPONENTS

QTY	DEVICE	DESCRIPTION
2	D82588	8 Mhz LAN Controller
2	PE64382	Dual Pulse Transformer Package
2	Printed Circuit Board	4x5 In. Kit Boards
2	Phone Jack	8 Position Phone Jack
1	Diskette	82588 Software Exerciser

LITERATURE

- StarLAN Application note
- LAN Components User's manual
- Pulse Transformer data sheet

2 82588 OVERVIEW

The 82588 is a highly integrated Local Area Network controller (LAN) that leads itself to cost sensitivity LAN applications. Because of its high integration the 82588 reduces component count which in turn reduces board space and development time. The 82588 provides most of the functions of the ISO physical and data link layers, including a CSMA/CD controller, two different collision detection mechanisms, and a data encoder/decoder supporting data transfers of up to 2 Mbs. As mentioned, one of the most important 82588 innovations is its on-chip collision detection mechanisms.

Two schemes are provided. First the code violation method defines a collision when a transition edge occurs outside of the region specified by NRZI or manchester encoding. Second, the bit comparison method compares the signature of a transmitted frame to the received one while listening to itself.

The 82588 goes beyond the basic provisions required by LAN controllers, by providing extensive programmability of its network and system interface parameters:

- Programmable station address length.
- Programmable station priority.
- Programmable interframe spacing.
- Programmable slot time.
- Programmable encoding scheme (manchester, NRZI, etc.).

The 82588 provides a 8 bit data path, and makes use of either one or two DMA channels for its data transfers from and to memory. Optimal utilization of system bus bandwidth is possible due to the two separated 16 byte Fifos available on chip.

Finally the 82588 provides a rich set of diagnostic and network management functions that allow the designer to minimize debug time and maintain top network efficiency.

3 STARLAN OVERVIEW

The proliferation of data outside a centrally controlled environment has seen a huge growth with the massive use of personal computers. Together with this growth has come the necessity of being able to share and manage this distributed data. Local networks (Ethernet) have emerged as high performance (10 Mbs) and expensive schemes (thousands of dollars per node), in a lot of cases affordable only to the largest corporations. Today we are seeing a trend to promote standards that will enable inexpensive link of personal computers (hundreds of dollars range), while maintaining the high functionality found in more expensive networks.

StarLAN, which was originally announced by ATT, and is now being studied by the IEEE 802.3 study group, is one of the most promising candidates to bring this wish to reality. StarLAN most significant selling point is the ability to use commonly available phone wires, often already present in most buildings. Cost of the wiring has been shown to be one of the most significant factors in any networking scheme, it has been shown by ATT that in many cases offices have enough spare wiring (25 pairs) to enable this kind of service.

For more detailed information about StarLAN, please refer to ap note AP-236, Implementing StarLAN with the Intel 82588 Controller (this document is part of your kit package).

4 HARDWARE INTERFACES

The StarLAN board has basically two major interfaces. On its bus side it is designed to interface to the IBM/PC bus (Refer to the IBM/PC technical reference manual for complete details), whereas in the network side it interfaces through an 8 pin phone jack to the StarLAN twisted pair wiring. The StarLAN board uses an internal 8 Mhz crystal oscillator, for both system and serial interfaces. This is the maximum bus interface speed, enabling the StarLAN board to be plugged not only into the IBM/PC but also in the various different PC versions (AT, XT, etc.) and its compatibles. The StarLAN board uses sixteen address spaces of the PC I/O space, it needs either one or two DMA channels and one INTERRUPT line.

For the following discussions refer to the board schematics at appendix C.

4.1 Bus Interfaced

As described the StarLAN board interfaces to the IBM/PC bus, it uses an internal 8 Mhz clock providing a 4 Mbs maximum DMA speed. The StarLAN board is designed to be assessed through two I/O ports. It uses an 8 bit data bus which is buffered for minimum load.

IBM/PC BUS

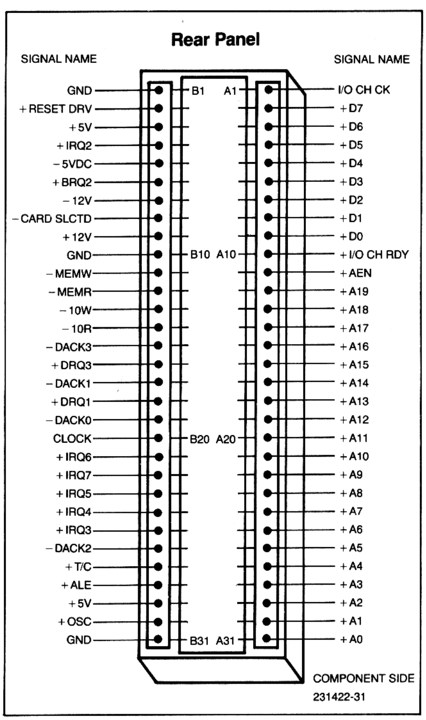
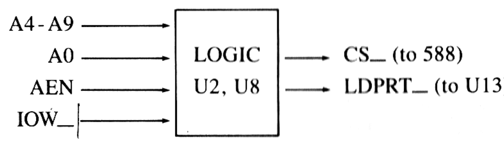


Figure 1. I/O Channel Diagram

4.1.1 Register Access and Data Bus Interface

The CPU accesses the StarLAN board through 2 I/O address windows. Address 300H is used to access the 82588 for commands and status, address 301H accesses an on board port that enables the various interrupt and DMA lines. Even though only two addresses are needed, the StarLAN board uses all the 16 addresses spaces from 300H to 30FH. This was done to keep simplicity and minimum component count. Registers address decoding is done using a PAL (16L8) and an external NAND gate (U8).

REGISTER ACCESS



FORMAT OF FOLLOWING EQUATIONS WILL BE ACCORDING TO THE FOLLOWING SPECIFICATIONS:

- ! INVERT
- _ SIGNAL ACTIVE LOW
- & LOGIC AND
- \+ LOGIC OR

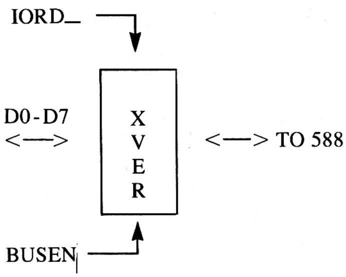
A9NANDA8 = !(A9 & A8) (DONE WITH U2)

CS_ = ! (!AEN & !A9NANDA8 & !A7 & !A6 & !A5 & !A4 & !A0)

LDPORT_ = ! (!AEN & !A9NANDA8 & !A7 & !A6 & !A5 & !A4 & A0 & !IOWR_)

The signal CS decodes address 300H, it is only active when AEN is inactive meaning CPU and not DMA cycles. LDPORT has exactly the same logic for address 301H, but it is only active during I/O write cycles. The I/O port sitting on address 301H is write only.

DATA BUS INTERFACE



BUSEN_ = DACK1_ & DACK2_ & (! (!AEN & !A9NANDA8 & !A7 & !A6 & !A5 & !A4));

The Bus transceiver is enabled if: A DMA access is taking place, or I/O ports 300H to 30FH are being accessed.

4.1.2 Control Port

As mentioned the StarLAN board has a 4-bit write only control port. The purpose of this port is to selectively enable the DMA and INTERRUPT request lines. Also it can completely disable the transmitter.

Control Port Definition

ENDRQ1	ENDRQ3	ENINTER	TXEN

ENDRQ1, ENDRQ2 : "1" Enable DMA requests.
ENINTER : "1" Enable INTERRUPT request.
TXEN : "1" Enable the transmitter.

On power up all bits default to "0".

4.1.3 Clock Generation

The 82588 requires two clocks for operation. The system clock and the serial clock. The serial clock can be generated on chip by putting a crystal across X1 and X2 pins. Alternatively, an externally generated clock can be fed in at pin X1 (with X2 left open). In both cases, the frequency must be either 8 or 16 times (sampling factor) the desired bit rate. For StarLAN, 8 or 16 MHz are the correct values to generate 1 Mb/s data rate. A configuration parameter is used to tell the 82588 what the sampling factor is. An externally supplied clock must have MOS levels (0.6 V-3.9 V).

The system clock has to be supplied externally. It can be up to 8 MHz. This clock runs the parallel side of the 82588. Its frequency does not have any impact on the read and write access times but on the rate at which data can be transferred to and from the 82588 (Maximum DMA rate is one byte every two system clocks). This clock doesn't require MOS levels.

The I/O channel of the IBM PC supplies a 4.77 MHz signal of 33% duty cycle. This signal could be used as a system clock. It was decided, however, to generate a separate clock on the StarLAN board to be independent of the I/O channel clock so that this board can also be used in other IBM PCs and also in some other compatibles. The 8 MHz system clock is generated using a DIP OSCILLATOR which have the required 50 ppm tolerance to meet StarLAN. This clock is converted to MOS levels by a 74HCT00 and fed into both the system and serial clock inputs.

4.1.4 DMA Interface

The 82588 requires either one or two DMA channels for full operation. In this application, one channel is dedicated for reception and the other is used for transmissions and the other commands. Use of only one DMA channel is possible but may require more complex software, also some RX frames may be lost during switches of the DMA channel from the receiver to the transmitter (Those frames will be recovered by higher layers of the protocol). Using only one DMA channel will limit the 82588 loopback functionality. Therefore the recommendation is to operate with two DMA channels if available.

The IBM PC system board has one 8237A DMA controller. Channel 0 is used for doing the refresh of DRAMs. Channels 1, 2 and 3 are available for add-on boards on the I/O Channel. The floppy disk controller board uses the DMA channel 2 leaving exactly two channels (1 and 3) for the 82588. The situation is worse if the IBM PC/XT is used, since it uses channel 3 for the Winchester hard disk leaving just the channel 1 for the 82588. On the other hand, the IBM PC/AT has 5 free DMA channels. We will assume that 8237A DMA channels 1 and 3 are available for the 82588 as in the case of the IBM PC (The StarLAN board was checked in the IBM AT and works properly).

Since the channel 0 of 8237A is used to do refresh of DRAMs all the channels should be operated in single byte transfer mode. In this mode, after every transfer the bus is granted to the current highest priority channel. In this way, no channel can hog the bus bandwidth and, more important, the refresh of DRAMs is assured every 15 microseconds since the refresh channel (number 0) has the highest priority. This mode of operation is very slow again after every transfer. Demand mode of operation is a lot more suitable to 82588 but it cannot be used because of the refresh requirements (For the IBM/AT the DEMAND MODE could be used, considering that the refresh is done outside the DMA controller, also the StarLAN board was designed to support this option).

Whenever the 82588 interfaces to the 8237A in the single transfer mode, there is a potential 8237A lock-up problem. The 82588 may deactivate its DMA request line (DREQ) before receiving an acknowledge from the DMA controller. This situation may happen during command abortions, or aborted receptions. The 8237A under those circumstances may lock-up. In order to solve this potential problem, an external logic must be used to insure that DREQ to the DMA controller is never deactivated before the acknowledge is received. This logic is implemented in the 16L8 PAL in the StarLAN board.

The 82588 DREQ lines are connected to the IBM/PC bus through tri-state buffers which are enabled by writing to I/O port 301H. This function enables the use of either one or two DMA channels and also the sharing of DMA channels with other adapter boards.

4.1.5 Interrupt Controller

The 82588 interrupts the CPU after the execution of a command or on reception of a frame. It uses the 8259A interrupt controller on the system board to interrupt the CPU. There are 6 interrupt request lines, IRQ2 to IRQ7, on the I/O channel. In fact, none of the lines is completely free for use. To add any new peripheral which uses a system board interrupt, this interrupt needs to have the capability to share the specific line, by driving the line with a tri-state driver. The 82588 StarLAN adapter board can optionally drive interrupt lines IRQ3, IRQ4 or IRQ5 (A 74LS125 driver is used).

4.2 Serial Link Interface

A typical StarLAN adapter board is connected to the twisted pair wiring using an extension cable (typically up to 8 meters — 25 ft.). One end of the cable plugs into the telephone modular jack on the StarLAN board and the other end into a modular jack in the wall. The twisted pair wiring starts at the modular jack in the wall and goes to the wiring closet. In the wiring closet, another telephone extension cable is used to connect to a StarLAN HUB. The transmitted signals from the 82588 reach the on-board telephone jack through a RS-422 driver with pulse shaping and a pulse transformer. The received signals from the telephone jack to the 82588 come through a pulse transformer, squelch circuit and a receive enable circuit.

4.2.1 Transmit Path

The single ended transmit signal on the TxD pin is converted to a differential signal and the rise and fall times are increased to 150 to 200 nanoseconds before feeding it to the pulse transformer (this pulse shaping is not a requirement, but proves to give good results). Am26LS30 is a RS-422 driver which converts the TxD signal to a differential signal. It also has slew rate control pins to increase to rise and fall times. A large rise and fall time reduces the possibility of crosstalk, interference and radiation. By the other hand a slower edge rate increases the jitter. In the StarLAN adapter card, the first approach was used. The 26LS30 converts a square pulse to a trapezoidal one. The filtering effect of the cable further adds to reduce the higher frequency components from the waveform so that on the cable the signal is almost sinusoidal. The pulse transformer is for DC isolation. The pulse transformer from Pulse Engineering — type PE 64382 — was used in this design. This is a dual transformer package which introduces an additional rise and fall time of about 70 - 100 ns on the signal, helping the former discussed waveshaping.

4.2.1.1 Idle Pattern Generation

StarLAN requires transmitters to generate an IDLE pattern after the last transmitted data bit. The IDLE pattern is defined to be a constant high level for 2-3 microseconds. The purpose of this pattern is to insure that receivers will decode properly the last transmitted data bits before signal decay. Currently the 82588 needs one external component to generate the IDLE. The operation principle is to have an external shift register (74LS164) that will kind of act as an envelope detector of the TXD line. Whenever the TXD line goes low (first preamble bit), the output of the shift register (third cell) will immediately go low, enabling the RS-422 driver, the shift register being clocked by TCLK will time the duration of the TXD high times. If the high time is more than 2 microseconds, meaning that the 82588 has gone idle, the transmitter will be disabled. Another piece of this logic is the ORing of the output of the shift register with TXEN signal which comes from the board control port. This signal completely disables the transmitter. The other purpose of this enable signal, is to make sure that after power-up, before the 82588 is configured, the RS-422 drivers won't be enabled (TCLK is not active before the configure command).

4.2.2 Receive Path

The signal coming from the HUB over the twisted pair wire is received on the StarLAN board through a 100 ohms line termination resistor and a pulse transformer. The pulse transformer is of the same type as for the transmit side and its function is dc isolation. The received signal which is differential and almost sinusoidal is fed to the AM26LS32 RS-422 receiver. As seen from the board schematics, the pulse transformer feeds two RS-422 receivers. The one on the bottom is for squelch filtering and the one above is the real receiver which does real zero crossing detection on the signal and regenerates a square digital waveform from the sinusoidal signal that is received. Proper zero crossing detection is very essential; if the edges of the regenerated signal are

16 - 8
14 - 6X2 = 12

not at zero crossings, the resulting signal may not be a proper Manchester encoded signal (self introduced jitter) even if the original signal is valid Manchester. The resistors in the lower receiver keep its differential inputs at a voltage difference of 600 mV. These bias resistors ensure that the output of the lower receiver remains high as long as the input signal is more than −600 mV. It is very important that the RxD pin remains HIGH (not LOW or floating) whenever the receive line is idle. A violation of this may cause the 82588 to lock-up on transmitting. Remember, that based on the signal on the RxD pin, the 82588 extracts information on the data being received, Carrier Sense and Collision Detect. This squelch of 600 mV keeps the idle line noise from getting to the 82588. The output of the squelch goes to a pulse stretcher which generates an envelope of the received frame. The envelope is a receive enable signal and is used to AND the signal from the real zero crossing receiver before feeding it to the RxD pin of the 82588.

4.3 Power Requirements

The StarLAN board requires a maximum of 1 ampere from the + 5 volts power supply.

5 BOARD ASSEMBLING

The StarLAN board is a relatively easy and simple to assemble board. Following carefully the assembling instructions is recommended to make sure the board is going to be functional without problems.

5.1 Parts List

REFERENCE	MFR. PART NUMBER	DESCRIPTION
16 U1	74LS191 .49	4 Bit Counter
14 U2	74LS04 .16	Hex-Inverter
14 U3	Crystek 8.000 MHz CC0-010-50	8 MHz, 50 ppm Crystal Oscillator
14 U4	74HCT00 .69	Quad- CMOS NAND Gates
20 U5	16L8	20 Pin PAL
16 U6	26LS32	RS-422 Receiver
16 U7	26LS30	RS-422 Transmitter
14 U8	74LS32 .15	Quad OR-Gates
14 U9	74LS125 .39	Quad Tri-State Buffers
16 U10	74LS175 .99	Quad D-FF With Inv. Outputs
14 U11	74LS164 .49	Serial-In, Parallel Out S-R
28 U12	82588	LAN Controller
20 U13	74LS245	8 Bit Transceiver
T1	PE-64382	Pulse Transformer
PJ1	—	Phone Jack
R1,R2,R3	2.2K Ohm, 1/4 W, 5%	Resistors
R4	100 Ohm, 1/4 W, 5%	Resistor
R5,R6	300 Ohm, 1/4 W, 5%	Resistors
C1,..,C4	.047 μF	Decoupling Capacitors
C5	10 μF, 50V	Decoupling Capacitor
C6,..,C13	.047 μF	Decoupling Capacitors
C14	10 μF, 50V	Decoupling Capacitor

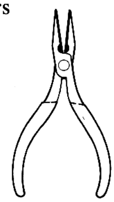
5.2 Getting Organized

Before you get started, it's a good idea to plan and organize your work space. Be sure you have ample room to accommodate this manual, lying open, and the circuit board, tools and your soldering pencil. If you don't have the traditional plastic, compartmented parts boxes, a muffin pan, an egg carton or some small boxes or jars can be used for parts sorting. It also might prove helpful to write the part values on small cards as you identify each part and place the card with the part for quick identification. Be organized and arrange everything within easy reach and you'll do the job quickly and minimize your chances for an error.

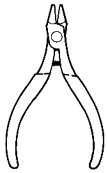
5.3 The Tools You Will Need

You will need the following tools and materials to assemble your kit:

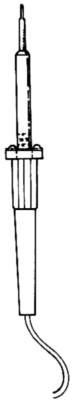
☐ Needle-nose pliers

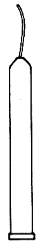


☐ Small diagonal wire cutters



☐ Soldering pencil. Not more than 30 watts with an extra-small diameter tip (1/16 inch)



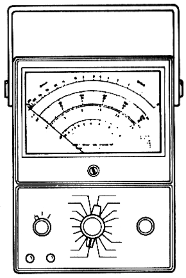


☐ Rosin-core solder. 60-40 (60% tin, 40% lead), small diameter (0.05 inch or less)

NOTE

Soldering paste or flux is not needed. The solder contains a sufficient amount of flux.

☐ Volt-Ohm-Milliameter



It also may prove useful to have a:

☐ Soldering aid with a small-tipped fork at one end and a reamer at the other. This tool will help to maneuver leads into holes and to manipulate small parts.

In the event that you happen to make an error and must remove a part that has been soldered from the circuit board, the following items will simplify the operation:

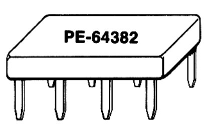
☐ Desoldering device, either the bulb variety (shown) or the pump variety.

☐ Length of copper braid ("solder wick") to draw solder out of a hole or to remove a solder bridge between circuits.

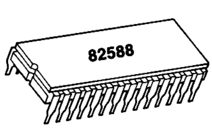
5.4 Getting the Necessary Parts

Provided with your kit:

☐ 2 Pulse Transformer, PE-64382

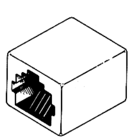


☐ 2 82588 LAN Controller



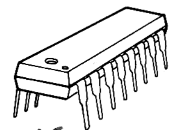
28 Pin Integrated Circuit Shown

☐ 2 Phone Jack



Additional parts to be acquired:

INTEGRATED CIRCUITS



.35

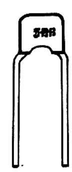
☐ 2 74LS04 .59
☐ 2 74LS191 .17
☐ 2 74HCT00
☐ 2 26LS30
☐ 2 26LS32
☐ 2 74LS32
☐ 2 16L8 (PAL) - 3.25 ea
☐ 2 74LS125 .49
☐ 2 74LS175 .49
☐ 2 74LS164
☐ 2 74LS245

89 59

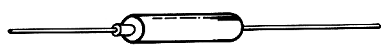
RESISTORS, 1/4 WATT, 5%

☐ 6 2.2K Ohms (red, red, red)
☐ 4 300 Ohms (orange, black, brown)
☐ 2 100 Ohms (brown, black, brown)

CAPACITORS

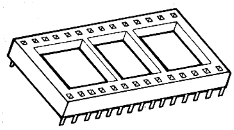


☐ 24 0.047 μF ceramic
☐ 4 5 pF (ceramic, dipped mica)



☐ 4 10 μF, electrolytic

INTEGRATED CIRCUIT SOCKETS



28 Pin Socket Shown

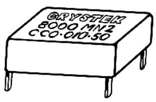
- ☐ 2 28 pin
- ☐ 2 20 pin

X 2 = (4)

12 - 14 pin

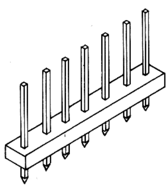
8 - 16 pin

CRYSTAL OSCILLATOR

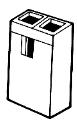


- ☐ 2 8 MHz, 50 ppm

JUMPER HEADERS



- ☐ 8 1 row x 3 positions



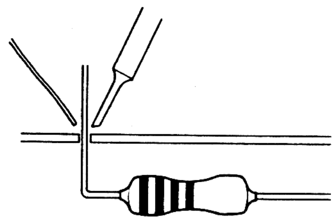
- ☐ 6 shorting plugs

5.5 Basic Soldering Techniques

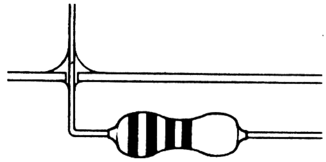
It can not be overstressed that the proper soldering pencil and solder are essential to good soldering. The soldering pencil (not soldering "iron" or soldering "gun") must be rated at no more than 30 watts, the tip must be small (1/16 inch or less), and small-diameter (0.05 inch) 60-40 solder must be used.

In addition to the right soldering equipment, a little care for the tip of the pencil certainly is worth mentioning. When soldering, the tip must be kept clean. If your pencil holder does not include a sponge and basin, dampen an old sponge with water and keep it on a plate or ashtray near the holder. When residue or excess solder builds up on the tip, draw the tip over the sponge a few times. When clean, "tin" the tip by melting a small amount of solder directly on the tip. A clean and properly tinned tip is very shiny.

If you are right-handed, hold the pencil in your right hand and, conversely, if you are left-handed, hold the pencil in your left hand. (Since most people have better control with their "natural" hand, the chances of slipping with the pencil and damaging the board or burning yourself are much less.) To solder a part, place the tip of the pencil on the pad and against the lead at the same time (this causes both the pad and lead to be heated evenly).

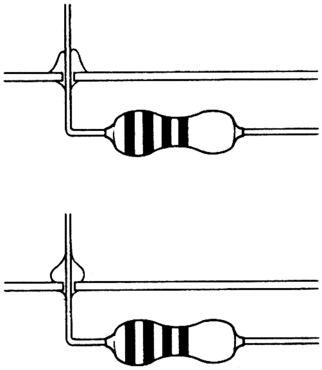


Apply solder from the side opposite the tip. Allow the solder to melt against the lead and pad and then remove both the solder and tip. Only a small amount of solder is needed for a good bond between the lead and pad. The following illustrations show a cross-section view of good and bad solder bonds or "joints." A good solder joint will be bright (shiny) and the solder around the lead and pad (referred to as the "fillet") will be distributed evenly. The shape of the fillet will range from being slightly concave to being slightly convex, and a small amount of solder usually will be drawn down through the hole and will be visible on the lead on the other side of the circuit board.



GOOD SOLDER JOINT

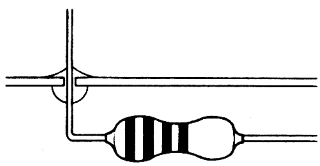
The bad soldering joints shown are typical of "cold" solder joints and usually are caused by not heating both the pad and lead uniformly or by allowing the solder to melt directly on the tip before the lead and pad have been heated properly.



BAD SOLDER JOINTS

Generally, a cold solder joint is dull gray in color and the bond at either the lead or pad will recede slightly.

Caution again must be emphasized in the selection of solder pencil wattage and tip size. Excessive (or prolonged) head can damage a part internally. When soldering, if the fillet between the pad and lead tends to "flatten out" and if an excessive amount of solder forms on the other side of the circuit board where the lead enters the hole, excessive heat may be indicated.



EXCESSIVE HEAT

5.6 Basic Assembly Techniques

Now that you have reviewed the basic soldering techniques, thoroughly familiarize yourself with the following sections on basic assembly techniques before you start to assemble your kit.

5.7 Circuit Board Layout

Pick up the circuit board and notice that one side has been silkscreened with component reference designations and position outlines, major circuit areas and other information. This side of the board will be referred to as the "top," and the unlabeled side will be referred to as the "bottom." Parts are inserted in the board from the top (over their outlines), the board is turned over, and the parts are soldered from the bottom.

5.8 Parts Orientation

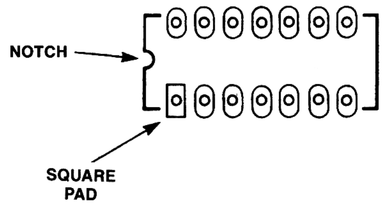
A number of the parts in your kit are "polarized." That is, they must be installed in a certain direction. Polarized parts include the following:

- Integrated Circuits
- Crystal Oscillator
- Electrolytic Capacitors
- Pulse Transformer

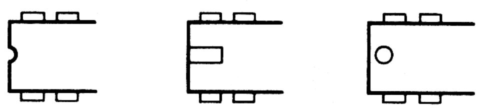
INTEGRATED CIRCUITS

When installing an integrated circuit into its designated circuit board location, always align pin 1 of the integrated circuit with pad 1 of the circuit board location.

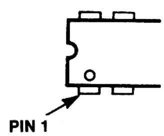
Looking at one of the silkscreened integrated circuit outlines on the top of the board, notice that one of the two outlines is "notched." Looking at the pad next to the notch notice that it is square-shaped and that the other pads have rounded corners. This is pad 1.



Now look at one of the small integrated circuits. One end of the circuit has a "notch." Depending on the manufacturer of the integrated circuit, the "notch" will look like one of the following:



Additionally, some manufacturers mark pin 1 with a small dot, dimple or bump.



Whether or not pin 1 is marked, it is always in the same position with respect to the notch.

CAUTION

Be very careful to install integrated circuits correctly. Once an integrated circuit has been soldered in place, the recommended way to remove the integrated circuit is to cut it out (which destroys the circuit).

CRYSTAL OSCILLATOR

Notice that crystal oscillators have three round corners and one square one. The square one is pad 1 and should fit into the square pad on the printed circuit board.

ELECTROLYTIC CAPACITORS

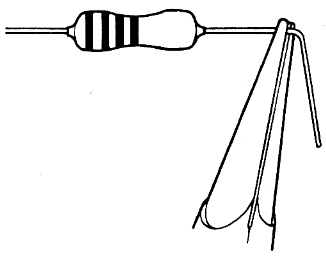
Electrolytic capacitors have a postive and a negative end. Looking at the leads of one of the electrolytic capacitors, the negative lead is in direct contact with the body of the capacitor, and the positive lead is insulated from the body and is noticeably larger at the point where the lead touches the body. On the circuit board, a small plus sign (+) is silkscreened near the positive hole to indicate the position of the capacitor's positive lead.

PULSE TRANSFORMER

The PE-64382 provided in your kit has pin numbers for each one of its pads. Pin 1 should fit into the square pad on the PCB.

5.9 Lead Bending

The leads of resistors, capacitors and electrolytic capacitors will have to be bent in order to insert the part into the circuit board. The correct way to bend a lead is to grip the lead with your needle-nose pliers and, using the pliers only to hold the lead (like a vise), bend the free-end of the lead with your other hand.

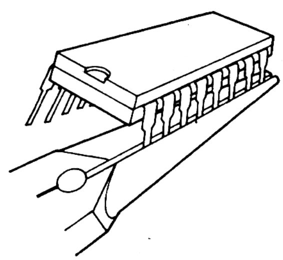


Never bend a lead with the pliers as the stress can fracture the part. Try to bend the leads so that when the part is inserted into the board, it is centered between the two pads. Also, when bending the leads of parts with values printed on them, bend the leads so that the value is visible when installed.

5.10 Inserting Parts

Before you insert any part in the circuit board, be sure that you have picked up the right part. When inserting the part, be sure that you put it in the right location and, if it is polarized, that you insert it in the proper direction. The top of the circuit board has been labeled with the part position, its reference number and, if it's an integrated circuit, with the integrated circuit part number. For small (two-lead) parts, a line has been silkscreened between its two associated pads, while with the larger (multi-pin) parts, the part outline has been silkscreened directly on the board.

Before inserting an integrated circuit or socket, check to be sure that all of the pins are straight. If necessary, straighten any bent pins with your needle-nose pliers. Occasionally, the pins of an integrated circuit may be spread too far apart to be inserted easily into the board. If this happens, carefully grip the pins on one side, in mass, with your needle-nose pliers and gently bend the pins inward.



If it still does not fit, bend the pins inward on the other side. After inserting an integrated circuit (or transistor or resistor pack), carefully check to be sure that:

1) The correct part has been inserted.
2) The part is facing in the right direction.
3) There are no bent pins (all pins must be visible from the bottom of the board).

Only when the above conditions have been checked carefully should the part be soldered in place. Remember that once an incorrectly installed integrated circuit has been soldered in place, the recommended method of removal is to cut out the integrated circuit.

NOTE

Although resistors can be inserted in either direction, it is good practice to insert all color-coded resistors so that the color bands are always read from the same direction (for instance, top-to-bottom or left-to-right).

5.11 Handling MOS Integrated Circuits

IMPORTANT

When instructed to install any of the large MOS integrated circuits, always observe the following precautions before you remove the skin-wrap and pick up the circuit:

1) Touch the black foam backing. Simply put your fingers on any exposed foam area.
2) Rub your fingers across the pads where you are going to insert the integrated circuit. If the circuit is to be inserted into a socket, *touch* the socket pins on the bottom of the board.
3) Avoid touching the pins of the integrated circuit as much as possible.
4) Avoid getting up and walking around (which could build up a static charge on your body) until you have inserted the integrated circuit in the board or into its socket.

5.12 Soldering Parts

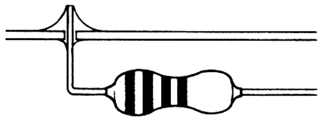
After a part or group of parts is inserted within an area, the board is turned over and the parts are soldered in place. To prevent parts from slipping or falling out when turning the board over, a piece of foam (about 1 foot square) can be placed over the parts to hold them in place or, if desired, parts can be taped down on the board with masking or similar type tape.

Small components (particularly the tiny capacitors) can be "tack" soldered (applying just enough solder to form a bond between one of the leads and its pad) on the top of the board. Once tacked in place, the board can be turned over, and the leads can be soldered. When soldering from the bottom, be sure to solder the "untacked" lead first.

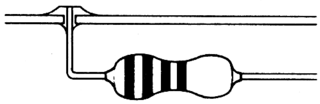
When soldering any of the multi-pin parts, first solder the pins at two diagonally opposite corners and then check to be sure that the part is all the way down on the board. If the part has slipped and is not down completely, remelt both solder joints while lightly pressing down on the component. Solder the remaining pins and then resolder the first two pins.

5.13 Clipping Leads

After soldering, the leads of most of the small parts must be clipped using diagonal wire cutters. Care should be taken to avoid clipping the solder fillet (which could possibly "fracture" the solder joint). When properly soldered and clipped, a lead should extend from 1/16 to 1/8 of an inch above the board. The following examples picture correct and improper lead clipping.



CORRECT



IMPROPER

Integrated circuit pins should not be clipped (the shock from clipping can cause an internal fracture).

WARNING

To avoid eye injury when clipping leads, hold the lead end so it can't fly into your face.

5.14 Removing Parts

Although the assembly instructions have been prepared carefully, the possibility of human error always exists. If you should happen to install a part incorrectly, it must be removed. Depending on the part itself, it either can be removed and reused or it should be cut out and replaced.

Small (two-lead) parts usually can be removed successfully from the board with little danger of damage, provided that the proper tools are used and care is taken. To remove a small part, find its leads on the bottom of the board. With your soldering pencil in one hand and the desoldering device in the other, heat one of the joints. When the solder melts, quickly move the pencil out of the way and remove the solder from the joint with the desoldering device. If done successfully, the hole in the pad will be visible. If solder still remains in the hole, first add additional solder to the joint before repeating the operation. After the solder has been removed from both joints, grasp each lead with your needle-nose pliers (from the bottom of the board) and wiggle the lead back and forth in order to break it loose from the hole. When both leads are free, remove the part. If a lead will not break free from its hole, resolder the lead and repeat the solder extraction operation.

While it is possible to remove an integrated circuit from the board without damaging the part, this practice is *not* recommended as the possibility of board damage (lifted

pads or traces) is extremely high. When an integrated circuit must be removed, it is recommended that the circuit be clipped out and the pins then be removed individually. To remove an integrated circuit, follow the steps outlined below.

1) Using your diagonal wire cutters and working from the top of the board, carefully clip each lead of the integrated circuit near where the pin enters the case.
2) While still working from the top of the board, carefully remove each individual lead using your soldering pencil and needle-nose pliers.
3) With your soldering pencil and solder, fill each hole with additional solder. (This will make it easier to remove all of the solder from the hole.)
4) Remove the solder from each hole using desoldering device or solder wick. If solder still remains in the hold, add more solder and repeat this step.
5) Install the correct replacement integrated circuit and make sure that pin 1 is oriented correctly.
6) Turn the board over and, checking first to be sure that all of the pins are visible, solder the circuit in place.

5.15 Assembly Steps

As you begin the assembly procedures, you will notice a small box to the left of each instruction or "step." AS YOU COMPLETE EACH STEP, PUT A CHECKMARK IN THE BOX! Remember that your goal is to build a working board, not to see how fast you can install the parts. Take your time and work carefully.

Let's get started by inserting all the passive elements. Starting by those will decrease the possibility of damaging ICs due to mishandling.

- ☐ Insert all decoupling capacitors (0.047 μF) C1, C2, C3, C4, C6, C7, C8, C9, C10, C11, C12, C13. The capacitors, as any other components, should be inserted in the component side of the board (Side where the component numbers are written).
- ☐ Insert 5 pF capacitors in C15, C16 places respectively.
- ☐ Insert electrolytic capacitors C5, C14. Make sure the polarity is correct by matching the capacitor "+" sign to the board one.
- ☐ Solder all the capacitors in place and clip their leads.
- ☐ Insert 2.2K resistors (red, red, red) in R1, R2, R3 places respectively.
- ☐ Insert 100 ohms resistor (brown, black, brown) into R4 place.
- ☐ Insert 300 ohms resistors (orange, black, brown) into R5, R6 places.
- ☐ Solder all the resistors in place and clip their leads.
- ☐ Insert jumper headers into W1, W2, W3, W4, W5 jumper posts.
- ☐ Solder all jumpers in place.
- ☐ Insert the 20 pin integrated circuit socket into U5. Notice that while sockets can be inserted in either direction, some sockets are marked with a pin 1 reference. When marked (usually with a "notch" or "sliced" corner), socket pin 1 should be aligned with the square pad.
- ☐ Insert the 28 pin IC socket into U12.
- ☐ Check to insure that all the socket pins come through the board and solder both U5, U12 sockets in place.
- ☐ Insert the PE-64382 pulse transformer into T1. Make sure pin marked 1 is aligned with the square pad.
- ☐ Solder the pulse transformer in place and clip its leads.
- ☐ Insert the crystal oscillator into U3. Make sure that the SQUARE side of the crystal oscillator is aligned to the square pad.
- ☐ Solder the oscillator in place and clip its leads.
- ☐ Insert 74LS191 IC at U1.
- ☐ Insert 74LS04 IC at U2.
- ☐ Insert 74HCT00 IC at U4.
- ☐ Check the orientation of the previous inserted three ICs. The notch of each should be to your left and pin 1 aligned to the square pad.
- ☐ Solder the 3 ICs in place. Do NOT clip the pins.
- ☐ Insert 26LS32 IC at U6.
- ☐ Insert 26LS30 IC at U7.
- ☐ Insert 74LS32 IC at U8.
- ☐ Insert 74LS125 IC at U9.
- ☐ Insert 74LS175 IC at U10.
- ☐ Once again check the previous components orientation and solder the ICs in place.
- ☐ Insert 74LS164 IC at U11.
- ☐ Insert 74LS245 IC at U13.
- ☐ Solder the previous two ICs in place.
- ☐ Insert the phone jack at PJ1 place.
- ☐ Solder the phone jack in place.

☐ Program the 16L8 PAL component by using the following equations.

```
PALSTAR DEVICE 'P16L8' ;

A9nandA8,A7,A6,A5,A4                    PIN 1,2,3,4,5;
IOWR_                                   PIN 6;
A0                                      PIN 7;
AEN                                     PIN 8;
REQ1                                    PIN 9;
REQ0                                    PIN 11;
LDPORT_                                 PIN 12;
RESET                                   PIN 13;
DACK3_                                  PIN 14;
DACK1_                                  PIN 15;
DREQ3                                   PIN 16;
DREQ1                                   PIN 17;
BUSEN_                                  PIN 19;
CS_                                     PIN 18;
VCC                                     PIN 20;
GND                                     PIN 10;

EQUATIONS
CS_ = ! ( !AEN & !A9nandA8 & !A7 & !A6 & !A5 & !A4 & !A0 ) ;

LDPORT_ = ! ( !AEN & !A9nandA8 & !A7 & !A6 & !A5 & !A4 & A0 & !IOWR_ ) ;

DREQ1 = ! ( (!REQ0 & !DACK1_ ) # ( !REQ0 & !DREQ1 ) # RESET ) ;

DREQ3 = ! ( ( !REQ1 & !DACK3_ ) # ( !REQ1 & !DREQ3 ) # RESET ) ;

BUSEN_ = DACK1_ & DACK3_ & (! ( !AEN & !A9nandA8 & !A7 & !A6 & !A5 & !A4)) ;

END STARPAL
```

☐ Insert the programmed PAL at U5 socket.

☐ Insert the 82588 at U13 socket. WARNING: The 82588 is a MOS device and by such should be properly handled (refer to section 5-11).

Congratulations: you have successfully completed the assembly of one of your Kit boards. Before you go into the testing phase, a second board must be assembled. Please repeat once again exactly the same procedure for the second board. At this stage you will have more experience in assembling the second board. You are recommended to once again take your time and work carefully. Avoid short-cuts and follow the assembly instructions.

6 JUMPERS CONFIGURATION

Now that all of the electrical parts have been installed, a few of the shorting jumpers must be installed. Three shorting jumpers will be required for proper operation.

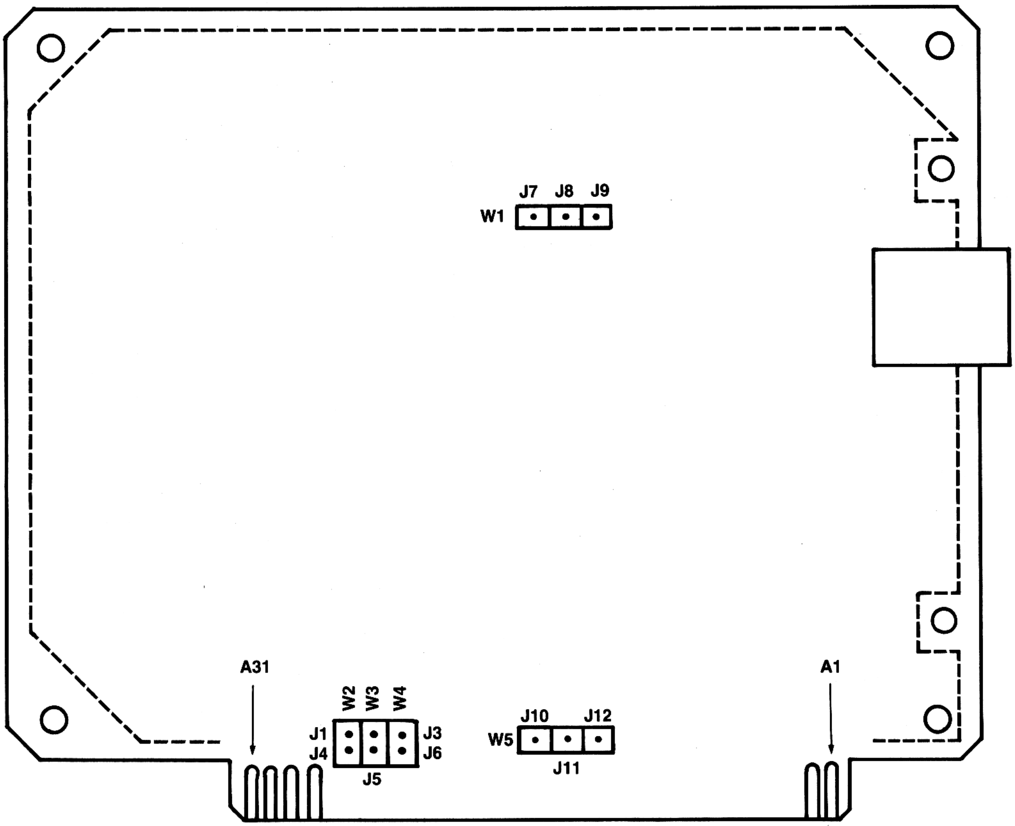


Figure 2

Default operation (supported by the EXERCISER software)

- Insert jumper in W1 shorting J7, J8.
- Insert jumper in W4 shorting J3, J6.
- Insert jumper in W5 shorting J11, J12.

The previous configuration selects two DMA channels usage, and interrupt request line INTR5

Other possible configurations:

Configuration	**Jumpers**	
One DMA channel usage	W1, J8-J9	SHORTED
	W5, J10-J11	SHORTED
Using INTR4 line	W3, J2-J5	SHORTED
Using INTR3 line	W2, J1-J4	SHORTED

7 DESIGN KIT EXERCISING SOFTWARE

Now that the boards are assembled, you are ready to system test them. Provided in your kit is the 82588 EXERCISER software which will enable you easy access to the board hardware and functionality.

7.1 Purpose

The main purpose of the 82588 EXERCISER software is to enable you to get acquainted with StarLAN and the 82588 on a hands on experience basis. The EXERCISER enables you to generate and monitor traffic on the link as well as to exercise the 82588 functionality. The EXERCISER was designed to be an easy to use tool. Most of the commands are menu driven and self explanatory. Following is a brief explanation of the software functionality and commands.

7.2 Screen Fields

The 82588 EXERCISER divides the screen in various fields for data displaying and prompting.

82588 EXERCISER SOFTWARE

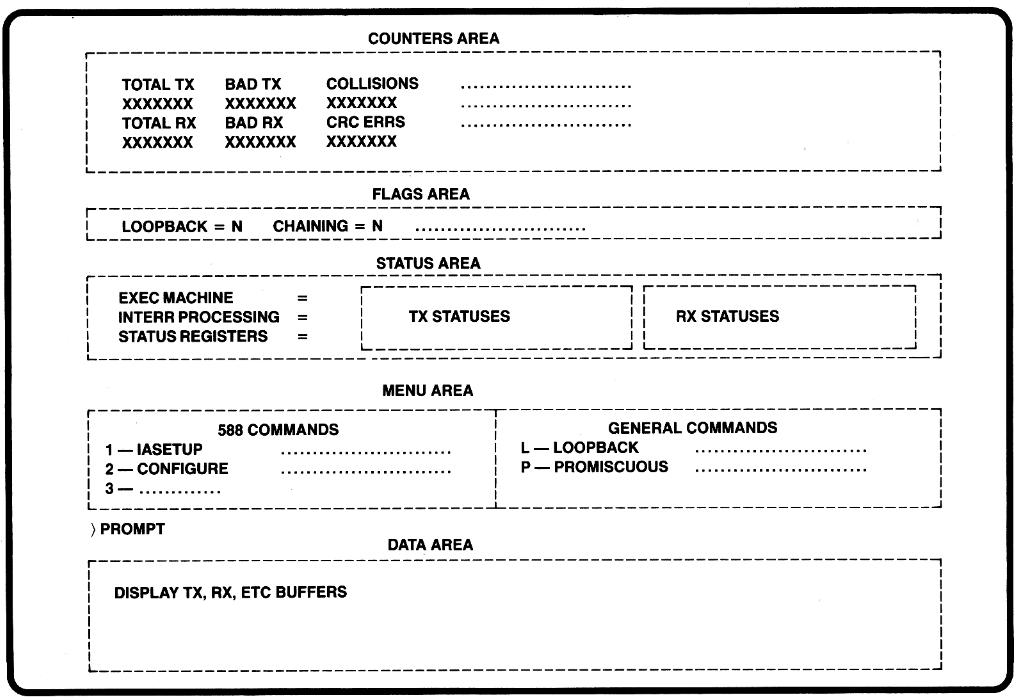


Figure 3

1. COUNTERS AREA

This is the area where statistics of executed transmissions and receptions are updated. The counters are hexadecimal ones.

Transmit Counters

- **TOTAL TX:** Indicates the total number of transmitted frames. It includes successful and unsuccessful ones, as well as retransmissions due to collisions.
- **BAD TX:** Indicates the number of unsuccessful transmissions.
- **COLLIS.:** Indicates the total number of collisions. Every transmitted or retransmitted frame that collides will increment this counter.
- **LOST CRS:** Indicates a LOSS OF CARRIER SENSE during transmissions. LOST CRS indicates that the transmitting station is not hearing its own carrier back.
- **UNDERUN:** Indicates data underruns during transmissions.
- **DEFERS:** Indicates number of frames that DEFERRED during their transmission.
- **LOST CTS:** Indicates LOST CLEAR TO SEND during transmissions.

Receive Counters

- **TOTAL RX:** Indicates total number of received frames. Includes good and bad ones.
- **BAD RX:** Indicates number of bad received frames.
- **CRC ERRS:** Indicates received frames with a CRC error.
- **OVERUN:** Indicates received frames with a data overrun condition.
- **SRT FRMS:** Indicates total number of short frames received.
- **ALG ERRS:** Indicates received frames with an alignment error.
- **NO EOF:** Indicates received frames that didn't have a CLOSING FLAG. This condition applies only for the BITSTUFFING mode of operation.

NOTE: For more information on each one of the transmit and receive statuses refer to the 82588 REFERENCE MANUAL (Part of the LAN components user's manual in your kit documentation).

2. FLAGS AREA

This area reports status of four flags available in the EXERCISER software:

- **LOOPBACK:** Indicates which of the loopback modes is active. N: for no loopback mode. I: for internal loopback. X: for external loopback.
- **CHAINING:** Indicates if chaining mode is currently active. N: for no chaining. C: for chaining active.
- **PROMISCUOUS:** Indicates if promiscuous mode is currently active. N: not active. P: for promiscuous active.
- **REPEAT:** Indicates if repeat mode is active. Repeat mode is used with the purpose of executing 82588 commands in loop. N: not active. R: currently in a command execution loop.

3. STATUS AREA

This area updates the user on commands/receptions execution. The area is divided in three sections:

- **EXEC MACHINE:** Indicates commands issued by the EXERCISER software to the 82588; e.g., IASETUP ISSUED, TX ISSUED, etc. This field indicates only command issuing, not completion.
- **INTERR PROCESSING:** Indicates results of 82588 interrupt processing. Interrupts are issued on command completions, and on new received frames. All the command interrupts are reported on the COMMAND/TX STATUS area, the receptions interrupts in the RX STATUS area.
- **STATUS REGISTERS:** Indicate the contents of the status registers (ST0, ST1, ST2, ST3) correspondent to the INTERR PROCESSING reporting.

4. MENU AREA

This is the area where the EXERCISER commands are displayed. Basically two kinds of commands exist: 82588 commands, and general commands. The 82588 commands are hexadecimal numbers (see 82588 REFERENCE MANUAL). General commands are one letter identifiers.

5. DATA AREA

General purpose data area. It is used for: Buffers display/modify, prompts, etc. This area is a window and will automatically scroll if needed.

7.3 Commands

As described before 82588 commands are hexadecimal numbers and completely match the 82588 command specifications. Those commands assume a buffer is already prepared in memory with necessary parameters/data for the respective command. The 82588 EXERCISER will automatically program the DMA and invoke the command. Changing of parameters/data in the buffers is possible by using the "Q" (alter) command described later.

82588 COMMANDS SEQUENCE:

1. Use "Q" commands to modify buffer if needed.
2. Press "R" key if command needs to be executed in loop (see repeat command).
3. Issue 82588 desired command by pressing 0,1,2,....,F
4. Command status is reported in the status area.

GENERAL COMMANDS

1) L — loopback command:

Enables selecting: No loopback, Internal loopback, External loopback modes. Choosing the required mode is done by pressing the "L" key as many times as needed. The mode will toggle cyclically between "N", "L", "X". No, Internal, and External loopback modes respectively.

2) P — Promiscuous mode command:

Enables choosing the promiscuous mode of operation. As before, pressing the "P" key toggles back and forth between promiscuous and normal modes.

3) O — Chaining mode command:

Enables choosing the receive buffer chaining mode. Pressing of the "O" key toggles between the normal and chaining modes.

4) I — init command:

This command will invoke the EXERCISER initialization procedure. All modes, buffers are initialized to their default values.

5) R — repeat command:

This command is used to enable loop execution of various commands. This command is used as a prefix for the other execution commands.

Example: Executing transmissions in loop:

press "R", press "4"

The following commands can be executed in loop:

- Individual address set-up.
- Configure.
- Multicast.
- Tdr.
- Dump.
- Diagnose.
- Retransmit.

For any of the other commands the "R" prefix will be a NOP.

6) S — stat command:

Display the contents of the 82588 status registers.

7) M — Collisions statistics command.

This command will display a summary of the collisions statistics.

8) N — clear counter command:

This command clears all the statistics counters.

9) X — exam command:

This command is available to enable direct I/O port read and writes. The "X" command will put the user in the EXAM MODE where "I", "O" commands can be issued.

COMMAND FORMAT:

I/O WRITE: X
O ⟨port address⟩ ⟨8 bit data⟩ ⟨cr or space⟩

I/O READ: X
I ⟨port address⟩ ⟨cr or space⟩ DISPLAYED DATA

10) K — display buffers command:

This command is used to display buffer contents. The command polls for the required buffer:

I — Individual address buffer.
C — Configure buffer.
M — Multicast buffer.
T — Transmit buffer.
D — Dump buffer.
R — Receive buffer.

When displaying transmit or receive buffers, you will be polled for either and hexadecimal or ASCII output.

When displaying receive buffers, you will be polled for the specific buffer. Seven receive buffers are available (0-7).

Buffers format is exactly the same as specified in the 82588 REFERENCE MANUAL.

11) Q — alter command:

Enables altering of buffers. The buffers are the same ones available in the "K" command with addition to:

L — Transmit frame length buffer (two hexadecimal bytes)
B — Data chaining buffer length (two hexadecimal bytes)
X — Transmit buffer, ASCII data input only. This mode is useful for sending ASCII messages between two stations.

12) T — Display receive errors command:

This command will display a buffer that contains information on the last 20 bad received frames. Statuses bytes (st0, st1, st2, st3) for those specific frames will be displayed.

13) Z — exit command:

Returns program control to DOS.

COMMANDS EXAMPLES

- Figure 04: Using EXAM command to write to port 45h data AAh.
- Figure 05: Using ALTER command to display buffers data.
- Figure 06: After selecting "Q" (alter) command, "X" buffer (transmit, ASCII input) was selected.
- Figure 07: Displaying receive buffers using the "K" (display) command.
- Figure 08: Buffer 1 was selected and data is being displayed in the DATA BUFFER (hexadecimal data).
- Figure 09: Displaying collisions statistics with the "M" command.
- Figure 10: Displaying receive errors with the "T" command.

```
TOTAL TX  BAD  TX   COLLIS.   LOST CRS  UNDERUN   DEFERS    LOST CTS
00000000  00000000  00000000  00000000  00000000  00000000  00000000
TOTAL RX  BAD  RX   CRC ERRS  OVERUN    SRT FRMS  ALG ERRS  NO EOF
00000000  00000000  00000000  00000000  00000000  00000000  00000000
--------------------------------------------------------------------------------
LOOPBACK=N   CHAINING=N   PROMISCUOUS=N   REPEAT= N
--------------------------------------------------------------------------------
EXEC MACHINE      =
INTERR. PROCESSING=
STATUS  REGISTERS =
--------------------------------------------------------------------------------
MENU:     588 commands                 !      GENERAL commands
0 - nop     5 - tdr     A - rx_dis ! L - lpbck    S - stat      Q - alter
1 - iaset   6 - dump    B - rx_stp ! P - promis   M - coll      T - rx_err
2 - conf    7 - diag    C - retx   ! O - chain    N - clr_cnt   Z - exit
3 - mcast   8 - rx_en   D - abrt   ! I - init     X - exam
4 - txmit   9 - nop     E - rst    ! R - repeat   K - disp
--------------------------------------------------------------------------------
EXAMINE > 045 AA
```

Figure 4

```
TOTAL TX  BAD  TX   COLLIS.   LOST CRS  UNDERUN   DEFERS    LOST CTS
00000000  00000000  00000000  00000000  00000000  00000000  00000000
TOTAL RX  BAD  RX   CRC ERRS  OVERUN    SRT FRMS  ALG ERRS  NO EOF
00000000  00000000  00000000  00000000  00000000  00000000  00000000
--------------------------------------------------------------------------------
LOOPBACK=N   CHAINING=N   PROMISCUOUS=N   REPEAT= N
--------------------------------------------------------------------------------
EXEC MACHINE      =
INTERR. PROCESSING=
STATUS  REGISTERS =
--------------------------------------------------------------------------------
MENU:     588 commands              |      GENERAL commands
0 - nop      5 - tdr     A - rx_dis | L - lpbck    S - stat      Q - alter
1 - iaset    6 - dump    B - rx_stp | P - promis   M - coll      T - rx_err
2 - conf     7 - diag    C - retx   | O - chain    N - clr_cnt   Z - exit
3 - mcast    8 - rx_en   D - abrt   | I - init     X - exam
4 - txmit    9 - nop     E - rst    | R - repeat   K - disp
--------------------------------------------------------------------------------
>Q
 Select buffer: I, C, M, T, D, R, L, B, X >
```

Figure 5

```
TOTAL TX  BAD  TX   COLLIS.   LOST CRS  UNDERUN   DEFERS    LOST CTS
00000000  00000000  00000000  00000000  00000000  00000000  00000000
TOTAL RX  BAD  RX   CRC ERRS  OVERUN    SRT FRMS  ALG ERRS  NO EOF
00000000  00000000  00000000  00000000  00000000  00000000  00000000
--------------------------------------------------------------------------------
LOOPBACK=N   CHAINING=N   PROMISCUOUS=N   REPEAT= N
--------------------------------------------------------------------------------
EXEC MACHINE      =
INTERR. PROCESSING=
STATUS  REGISTERS =
--------------------------------------------------------------------------------
MENU:     588 commands              |      GENERAL commands
0 - nop      5 - tdr     A - rx_dis | L - lpbck    S - stat      Q - alter
1 - iaset    6 - dump    B - rx_stp | P - promis   M - coll      T - rx_err
2 - conf     7 - diag    C - retx   | O - chain    N - clr_cnt   Z - exit
3 - mcast    8 - rx_en   D - abrt   | I - init     X - exam
4 - txmit    9 - nop     E - rst    | R - repeat   K - disp
--------------------------------------------------------------------------------
>Q
ASCII INPUT > THIS AN EXAMPLE OF DATA INPUT FOR A TRANSMIT ASCII BUFFER.
THIS DATA WILL BE DEPOSIT IN THE TRANSMIT DATA BUFFER AREA AFTER THE FRAME LENGT
H AND THE DESTINATION AND SOURCE ADDRESS. THE FRAME LENGTH WILL NOT BE AUTOMATIC
ALLY MODIFIED. **
```

Figure 6

```
TOTAL TX  BAD  TX   COLLIS.   LOST CRS  UNDERUN   DEFERS    LOST CTS
00000000  00000000  00000000  00000000  00000000  00000000  00000000
TOTAL RX  BAD  RX   CRC ERRS  OVERUN    SRT FRMS  ALG ERRS  NO EOF
00000000  00000000  00000000  00000000  00000000  00000000  00000000
--------------------------------------------------------------------------------
LOOPBACK=N   CHAINING=N   PROMISCUOUS=N   REPEAT= N
--------------------------------------------------------------------------------
EXEC MACHINE       =
INTERR. PROCESSING=
STATUS  REGISTERS =
--------------------------------------------------------------------------------
MENU:      588 commands              |      GENERAL commands
0 - nop      5 - tdr    A - rx_dis | L - lpbck     S - stat      Q - alter
1 - iaset    6 - dump   B - rx_stp | P - promis    M - coll      T - rx_err
2 - conf     7 - diag   C - retx   | O - chain     N - clr_cnt   Z - exit
3 - mcast    8 - rx_en  D - abrt   | I - init      X - exam
4 - txmit    9 - nop    E - rst    | R - repeat    K - disp
--------------------------------------------------------------------------------
>K
 Select buffer: I, C, M, T, D, R  >R
 Select code : A-ascii, H-hex      >R
 Select buffer (0 -> 7, <space> for current )
```

Figure 7

```
TOTAL TX  BAD  TX   COLLIS.   LOST CRS  UNDERUN   DEFERS    LOST CTS
00000000  00000000  00000000  00000000  00000000  00000000  00000000
TOTAL RX  BAD  RX   CRC ERRS  OVERUN    SRT FRMS  ALG ERRS  NO EOF
00000000  00000000  00000000  00000000  00000000  00000000  00000000
--------------------------------------------------------------------------------
LOOPBACK=N   CHAINING=N   PROMISCUOUS=N   REPEAT= N
--------------------------------------------------------------------------------
EXEC MACHINE       =
INTERR  PROCESSING=
STATUS  REGISTERS =
--------------------------------------------------------------------------------
MENU:      588 commands              |      GENERAL commands
0 - nop      5 - tdr    A - rx_dis | L - lpbck     S - stat      Q - alter
1 - iaset    6 - dump   B - rx_stp | P - promis    M - coll      T - rx_err
2 - conf     7 - diag   C - retx   | O - chain     N - clr_cnt   Z - exit
3 - mcast    8 - rx_en  D - abrt   | I - init      X - exam
4 - txmit    9 - nop    E - rst    | R - repeat    K - disp
--------------------------------------------------------------------------------
     BUFFER :  1  STATUS: 00 00  BUFFERS CHAINED: 00
60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
more data ?, <cr> to finish :
```

Figure 8

```
  COLLISIONS SUMMARY
collision counter      -    01 -00000000
collision counter      -    02 -00000000
collision counter      -    03 -00000000
collision counter      -    04 -00000000
collision counter      -    05 -00000000
collision counter      -    06 -00000000
collision counter      -    07 -00000000
collision counter      -    08 -00000000
collision counter      -    09 -00000000
collision counter      -    0A -00000000
collision counter      -    0B -00000000
collision counter      -    0C -00000000
collision counter      -    0D -00000000
collision counter      -    0E -00000000
collision counter      -    0F -00000000

total, max collisions -         00000000
total  number of coll -         00000000
                          press any key to end
```

Figure 9

```
  20 LAST RX ERRORS SUMMARY
st0  st1  len0  len1
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00
 00   00   00   00   press any key to end
```

Figure 10

8 SYSTEM CONFIGURATIONS AND BOARD EXERCISING

Now that you are acquainted with the 82588 EXERCISER, you should be ready to start exercising your board's functionality. As described in the included AP-NOTE, StarLAN needs a HUB in order to connect stations to each other. Two options will be described in order to exercise your board's functionality. The first option is to use the 82588 in LOOPBACK mode. This option is the quickest and simplest way to get started,but it doesn't really exercise the StarLAN full functionality. The second and most recommended option is to acquire a commercially available HUB and build a real StarLAN network.

8.1 Loopback Operation

As described this is not the preferred mode of operation, but will allow you to get started with the 82588 functionality. The 82588 supports both internal and external loopback. For internal loopback, no other external hardware is needed. For external loopback, you will need an external cable and an external phone jack.

INTERNAL LOOPBACK

As said, you are all set for this mode of operation. This mode will test most of your board functionality but the analog interface.

Follow the following steps for board evaluation.

1) Power your PC off.
2) Plug the StarLAN board into an empty slot of the PC card cage.
3) Insert the 82588 EXERCISER diskette into drive A.

   NOTE: The 82588 EXERCISER software uses the ANSI.SYS device driver. This driver is included in your EXERCISER diskette; therefore it is important for the PC to boot with that disk into drive A.
4) Invoke the 82588 EXERCISER by typing "E588V1"

   On doing that the program will boot and the first 82588 CONFIGURE command will be executed. If a CONFIG. DONE status is not reported, your board is not functional. Refer to Appendix A in case of trouble.
5) In this mode as said you can only execute transmissions and receptions in the internal loopback mode. Execute the "L" command once to get into internal loopback mode.
6) Now every command can be executed according to the section 7.3 specifications.

EXTERNAL LOOPBACK MODE:

This mode's steps are exactly the same as internal loopback, the only difference is that you will need a twisted pair phone cord, and an extra phone jack. The phone jack must be wired such that the transmit pair is shortened to the receive pair.

(See Figure 11 for details.)

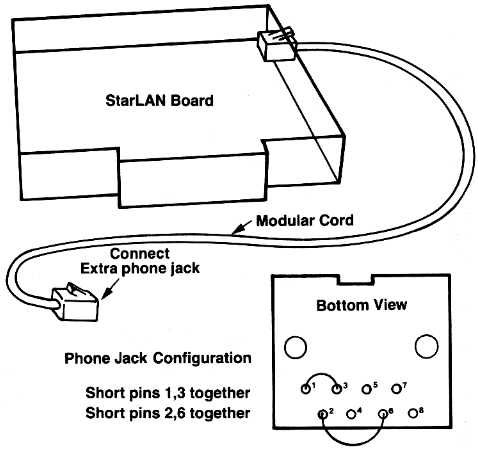


Figure 11 — External Loopback Configuration

The advantage of this mode compared to internal loopback, is that it indicates whether your transmitter analog interface is functional.

8.2 Using a Commercially Available HUB

This is the most recommended mode of operation. It will enable you not only to exercise the 82588 functionality but also exercise your boards on a "real" StarLAN environment. Figures 14, 15 show the network configuration using a HUB. Currently two HUBs are available in the market, from AT&T and from RETIX. Both HUBs should work with your StarLAN board. This design kit was specifically tested using the RETIX HB-12 device. RETIX offers two HUB types: the HB-6 which is a 6 port device, and HB-12 which is a 12 port one. HB-6 is a simpler one and does not include the retiming circuitry required by the 802.3 1BASE5 StarLAN draft. HB-12 is more complex including all the circuitry necessary to completely support the IEEE requirements. Either one of the HUBs is sufficient for getting started with the 82588 and StarLAN. As described, HB-6 doesn't include the retiming logic, and by such is limited to smaller topologies (up to 2 HUB levels), whereas HB-12 can go to the full StarLAN topology of 5 HUB levels.

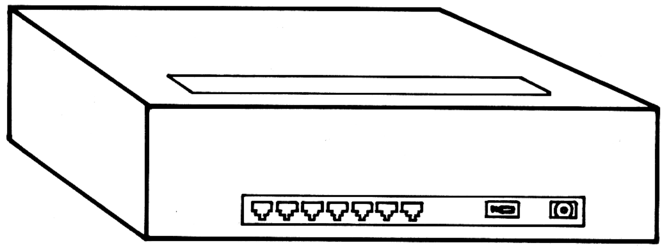
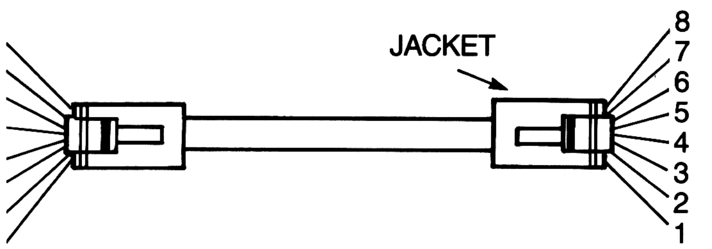


Figure 12. Retix StarSeries HB-6 Hub



BL	1	+ Upward Data
O	2	− Upward Data
BK	3	+ Downward Data
R	4	Not used
G	5	Not used
Y	6	− Downward Data
BN	7	Reserved
S	8	Reserved

Note: Stations and Hubs upper ports use contacts 1 and 2 for transmitting, and contacts 3 and 6 for receiving. Other Hub ports use 1 and 2 for receiving and 3 and 6 for transmitting.

Figure 13. 8-Pin Modular Plug Pin Usage

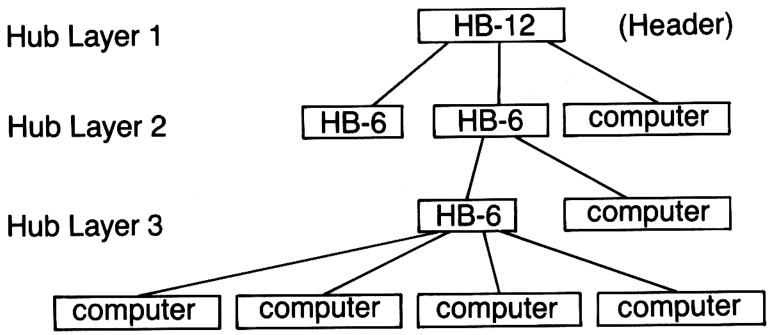


Figure 14. Sample 3-Layer StarLAN type Network

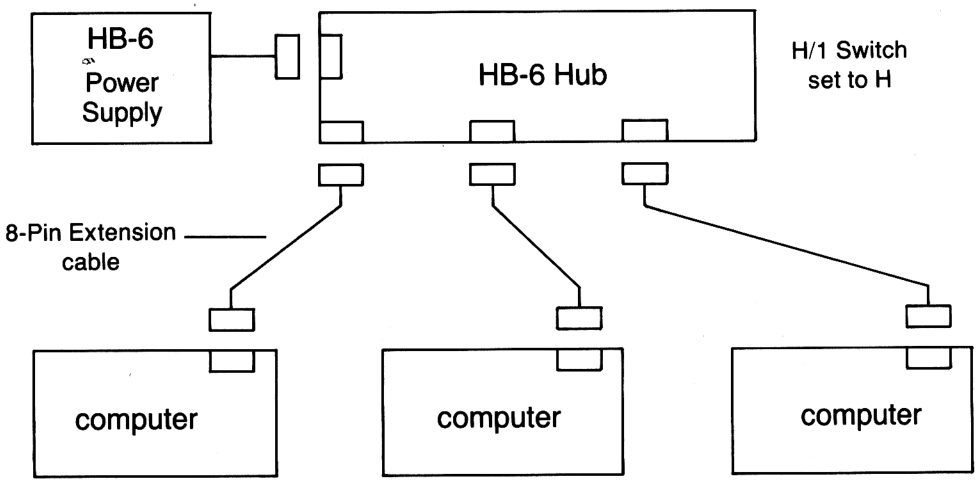


Figure 15. Example: Three Computer Network Hookup

RETIX HB-6, HB-12 FUNCTIONALITY:

Functions Common to Both HB-6, HB-12.

1) Repeater and signal amplification.
2) Collision detection per 1BASE5 specifications. Includes generation of the collision presence signal.
3) Noise suppression logic, which filters most electrical noise coming from adjacent cables.
4) HB-6 provides up to six ports.
5) HB-6 allows a maximum of two HUB levels.
6) Switch selectable as either Intermediate or Header HUBS.

Additional HB-12 Functionality

1) Retiming to full 1BASE5 specifications, allowing full topology to a maximum of 5 HUB levels.
2) Jabber Logic to disconnect when alloted transmit time is exceeded.
3) HB-12 provides 12 ports.

9 82588 EXERCISER EXAMPLES

The following examples will show usage of the StarLAN board in a stand-alone configuration (refer to section 8.1, Internal Loopback Mode). This is the simplest configuration to check your board functionality.

Load your EXERCISER program by invoking E588V1.EXE module.

- Figure 16: EXERCISER state immediately after program loading. As can be seen the first CONFIGURE command is automatically executed and the CONF. DONE status is displayed. If immediately after boot your program does not display the configure done status, your board may be malfunctioning (Refer to Appendix A for instructions).
- Figure 17: Press "1" to execute the individual address command.
- Figure 18: Press "6" to execute the dump command.
- Figure 19: Press "4" a few times to execute a few transmissions. As you can see in the status the transmissions are not successful, due to LOST CRS. This happens due to the fact that the board is not connected anywhere, and no Internal Loopback was chosen. (The transmitter is not hearing its own carrier.)
- Figure 20: Press "N" to clear the counters. Press "L" to set the Internal loopback mode. Press "R", "4" to execute multiple transmissions in loop. Press "space" to stop your loop. As can be seen the TOTAL TX counter increments indicating successful transmissions.
- Figure 21: Press "N" to clear counters. Press "P" to set the promiscuous mode. Press "8" to enable the receiver. Press "R", "4" to execute multiple transmissions. As can be seen the TOTAL TX, TOTAL RX counters are incrementing indicating successful transmissions and receptions (press "space" whenever you want to stop the loop).
- Figure 22: Use the command "Q" to enter an ASCII data buffer. Press "Q". Press "X". Enter your text. Press "CR" to finish.
- Execute a transmission and check the received data. Press "4". Press "K", "R", "A", "space." See that the received data matches the transmitted one. Notice that at the beginning of the buffer a few characters seem to be garbled. Those correspond to the Destination address, Source address, etc., which may not be ASCII data. For displaying additional data of the same buffer, press "space", press "CR" to go to the next buffer. Press "CR" twice to exit.
- Figure 23: Using the "Q" command to change the transmit buffer length, press "Q", enter "30", press "space", enter "00", press "CR". The transmit frame length is now changed to 30h (48 decimal) which is shorter than the default minimum frame length of 64 decimal.
- Figure 24: Execute multiple transmissions by pressing "R", "4". Note that all the received frames are bad and that the SHORT FRAMES counter is incrementing.

```
TOTAL TX  BAD  TX   COLLIS.   LOST CRS  UNDERUN   DEFERS    LOST CTS
00000000  00000000  00000000  00000000  00000000  00000000  00000000
TOTAL RX  BAD  RX   CRC ERRS  OVERUN    SRT FRMS  ALG ERRS  NO EOF
00000000  00000000  00000000  00000000  00000000  00000000  00000000
--------------------------------------------------------------------------------
LOOPBACK=N   CHAINING=N   PROMISCUOUS=N   REPEAT= N
--------------------------------------------------------------------------------
EXEC MACHINE      =
INTERR. PROCESSING=   CONF.   DONE
STATUS  REGISTERS =   B2 00 00 04
--------------------------------------------------------------------------------
MENU:      588 commands              |      GENERAL commands
0 - nop      5 - tdr     A - rx_dis | L - lpbck    S - stat     Q - alter
1 - iaset    6 - dump    B - rx_stp | P - promis   M - coll     T - rx_err
2 - conf     7 - diag    C - retx   | O - chain    N - clr_cnt  Z - exit
3 - mcast    8 - rx_en   D - abrt   | I - init     X - exam
4 - txmit    9 - nop     E - rst    | R - repeat   K - disp
--------------------------------------------------------------------------------
```

Figure 16

```
TOTAL TX  BAD  TX   COLLIS.   LOST CRS  UNDERUN   DEFERS    LOST CTS
00000000  00000000  00000000  00000000  00000000  00000000  00000000
TOTAL RX  BAD  RX   CRC ERRS  OVERUN    SRT FRMS  ALG ERRS  NO EOF
00000000  00000000  00000000  00000000  00000000  00000000  00000000
--------------------------------------------------------------------------------
LOOPBACK=N   CHAINING=N   PROMISCUOUS=N   REPEAT= N
--------------------------------------------------------------------------------
EXEC MACHINE      =    COMMAND  IA      ISSUED
INTERR. PROCESSING=  IA      DONE
STATUS  REGISTERS =  B1 00 00 04
--------------------------------------------------------------------------------
MENU:      588 commands              |      GENERAL commands
0 - nop      5 - tdr     A - rx_dis | L - lpbck    S - stat     Q - alter
1 - iaset    6 - dump    B - rx_stp | P - promis   M - coll     T - rx_err
2 - conf     7 - diag    C - retx   | O - chain    N - clr_cnt  Z - exit
3 - mcast    8 - rx_en   D - abrt   | I - init     X - exam
4 - txmit    9 - nop     E - rst    | R - repeat   K - disp
--------------------------------------------------------------------------------
>1
```

Figure 17

```
TOTAL TX  BAD  TX   COLLIS.   LOST CRS  UNDERUN   DEFERS    LOST CTS
00000000  00000000  00000000  00000000  00000000  00000000  00000000
TOTAL RX  BAD  RX   CRC ERRS  OVERUN    SRT FRMS  ALG ERRS  NO EOF
00000000  00000000  00000000  00000000  00000000  00000000  00000000
--------------------------------------------------------------------------------
LOOPBACK=N   CHAINING=N   PROMISCUOUS=N   REPEAT= N
--------------------------------------------------------------------------------
EXEC MACHINE       =     COMMAND  DUMP    ISSUED
INTERR. PROCESSING=   DUMP    DONE
STATUS  REGISTERS =   B6 00 00 04
--------------------------------------------------------------------------------
MENU:      588 commands             !      GENERAL commands
0 - nop      5 - tdr     A - rx_dis ! L - lpbck     S - stat      Q - alter
1 - iaset    6 - dump    B - rx_stp ! P - promis    M - coll      T - rx_err
2 - conf     7 - diag    C - retx   ! O - chain     N - clr_cnt   Z - exit
3 - mcast    8 - rx_en   D - abrt   ! I - init      X - exam
4 - txmit    9 - nop     E - rst    ! R - repeat    K - disp
--------------------------------------------------------------------------------
>6
```

Figure 18

```
TOTAL TX  BAD  TX   COLLIS.   LOST CRS  UNDERUN   DEFERS    LOST CTS
00000006  00000006  00000000  00000006  00000000  00000001  00000000
TOTAL RX  BAD  RX   CRC ERRS  OVERUN    SRT FRMS  ALG ERRS  NO EOF
00000000  00000000  00000000  00000000  00000000  00000000  00000000
--------------------------------------------------------------------------------
LOOPBACK=N   CHAINING=N   PROMISCUOUS=N   REPEAT= N
--------------------------------------------------------------------------------
EXEC MACHINE       =     COMMAND  TXMIT   ISSUED
INTERR. PROCESSING=   TXMIT   DONE
STATUS  REGISTERS =   B4 00 04 04
--------------------------------------------------------------------------------
MENU:      588 commands             !      GENERAL commands
0 - nop      5 - tdr     A - rx_dis ! L - lpbck     S - stat      Q - alter
1 - iaset    6 - dump    B - rx_stp ! P - promis    M - coll      T - rx_err
2 - conf     7 - diag    C - retx   ! O - chain     N - clr_cnt   Z - exit
3 - mcast    8 - rx_en   D - abrt   ! I - init      X - exam
4 - txmit    9 - nop     E - rst    ! R - repeat    K - disp
--------------------------------------------------------------------------------
>4
```

Figure 19

```
TOTAL TX  BAD  TX   COLLIS.   LOST CRS  UNDERUN   DEFERS    LOST CTS
000001FD  00000000  00000000  00000000  00000000  00000000  00000000
TOTAL RX  BAD  RX   CRC ERRS  OVERUN    SRT FRMS  ALG ERRS  NO EOF
00000000  00000000  00000000  00000000  00000000  00000000  00000000
-----------------------------------------------------------------------------
LOOPBACK=L   CHAINING=N   PROMISCUOUS=N   REPEAT= N
-----------------------------------------------------------------------------
EXEC MACHINE      =
INTERR. PROCESSING=   TXMIT   DONE
STATUS  REGISTERS =   B4 00 20 04
-----------------------------------------------------------------------------
MENU:      588 commands             |      GENERAL commands
0 - nop      5 - tdr     A - rx_dis | L - lpbck    S - stat      Q - alter
1 - iaset    6 - dump    B - rx_stp | P - promis   M - coll      T - rx_err
2 - conf     7 - diag    C - retx   | O - chain    N - clr_cnt   Z - exit
3 - mcast    8 - rx_en   D - abrt   | I - init     X - exam
4 - txmit    9 - nop     E - rst    | R - repeat   K - disp
-----------------------------------------------------------------------------
>
```

Figure 20

```
TOTAL TX  BAD  TX   COLLIS.   LOST CRS  UNDERUN   DEFERS    LOST CTS
00000114  00000000  00000000  00000000  00000000  00000000  00000000
TOTAL RX  BAD  RX   CRC ERRS  OVERUN    SRT FRMS  ALG ERRS  NO EOF
00000114  00000000  00000000  00000000  00000000  00000000  00000000
-----------------------------------------------------------------------------
LOOPBACK=L   CHAINING=N   PROMISCUOUS=P   REPEAT= N
-----------------------------------------------------------------------------
EXEC MACHINE      =    COMMAND  TXMIT   ISSUED
INTERR. PROCESSING=   TXMIT   DONE      NEW RX FRAME
STATUS  REGISTERS =   B4 00 20 24       C8 08 01 26
-----------------------------------------------------------------------------
MENU:      588 commands             |      GENERAL commands
0 - nop      5 - tdr     A - rx_dis | L - lpbck    S - stat      Q - alter
1 - iaset    6 - dump    B - rx_stp | P - promis   M - coll      T - rx_err
2 - conf     7 - diag    C - retx   | O - chain    N - clr_cnt   Z - exit
3 - mcast    8 - rx_en   D - abrt   | I - init     X - exam
4 - txmit    9 - nop     E - rst    | R - repeat   K - disp
-----------------------------------------------------------------------------
>4
```

Figure 21

```
 TOTAL TX  BAD  TX   COLLIS.   LOST CRS  UNDERUN   DEFERS    LOST CTS
 00000001  00000000  00000000  00000000  00000000  00000000  00000000
 TOTAL RX  BAD  RX   CRC ERRS  OVERUN    SRT FRMS  ALG ERRS  NO EOF
 00000001  00000000  00000000  00000000  00000000  00000000  00000000
--------------------------------------------------------------------------------
 LOOPBACK=L   CHAINING=N   PROMISCUOUS=P   REPEAT= N
--------------------------------------------------------------------------------
 EXEC MACHINE      =
 INTERR. PROCESSING=
 STATUS  REGISTERS =
--------------------------------------------------------------------------------
 MENU:     588 commands            |      GENERAL commands
 0 - nop     5 - tdr    A - rx_dis | L - lpbck    S - stat     Q - alter
 1 - iaset   6 - dump   B - rx_stp | P - promis   M - coll     T - rx_err
 2 - conf    7 - diag   C - retx   | O - chain    N - clr_cnt  Z - exit
 3 - mcast   8 - rx_en  D - abrt   | I - init     X - exam
 4 - txmit   9 - nop    E - rst    | R - repeat   K - disp
--------------------------------------------------------------------------------
>Q
ASCII INPUT > THIS IS AN EXAMPLE OF TRANSMISSION OF AN ASCII BUFFER. THIS BUFFER
 WAS INPUT USING THE "Q" COMMAND USING THE "X" OPTION. THIS BUFFER WILL BE TRANS
MITTED USING "4" THEN DISPLAYED USING "K"
```

Figure 22

```
 TOTAL TX  BAD  TX   COLLIS.   LOST CRS  UNDERUN   DEFERS    LOST CTS
 00000002  00000000  00000000  00000000  00000000  00000000  00000000
 TOTAL RX  BAD  RX   CRC ERRS  OVERUN    SRT FRMS  ALG ERRS  NO EOF
 00000002  00000000  00000000  00000000  00000000  00000000  00000000
--------------------------------------------------------------------------------
 LOOPBACK=L   CHAINING=N   PROMISCUOUS=P   REPEAT= N
--------------------------------------------------------------------------------
 EXEC MACHINE      =
 INTERR. PROCESSING=
 STATUS  REGISTERS =
--------------------------------------------------------------------------------
 MENU:     588 commands            |      GENERAL commands
 0 - nop     5 - tdr    A - rx_dis | L - lpbck    S - stat     Q - alter
 1 - iaset   6 - dump   B - rx_stp | P - promis   M - coll     T - rx_err
 2 - conf    7 - diag   C - retx   | O - chain    N - clr_cnt  Z - exit
 3 - mcast   8 - rx_en  D - abrt   | I - init     X - exam
 4 - txmit   9 - nop    E - rst    | R - repeat   K - disp
--------------------------------------------------------------------------------
     BUFFER :  6  STATUS: 00 A0  BUFFERS CHAINED: 00
??????██████THIS IS AN EXAMPLE OF TRANSMISSION OF AN ASCII BUFFER. THIS BUFFER W
AS INPUT USING THE "Q" COMMAND USING THE "X" OPTION. THIS BUFFER WILL BE TRANSMI
TTED USING "4" THEN DISPLAYED USING "K"
more buffers ?, <cr> to finish
```

Figure 23

```
TOTAL TX  BAD  TX   COLLIS.   LOST CRS  UNDERUN   DEFERS    LOST CTS
00000002  00000000  00000000  00000000  00000000  00000000  00000000
TOTAL RX  BAD  RX   CRC ERRS  OVERUN    SRT FRMS  ALG ERRS  NO EOF
00000002  00000000  00000000  00000000  00000000  00000000  00000000
--------------------------------------------------------------------------------
LOOPBACK=L   CHAINING=N   PROMISCUOUS=P   REPEAT= N
--------------------------------------------------------------------------------
EXEC MACHINE       =
INTERR. PROCESSING=
STATUS  REGISTERS =
--------------------------------------------------------------------------------
MENU:       588 commands               |       GENERAL commands
0 - nop      5 - tdr      A - rx_dis | L - lpbck     S - stat      Q - alter
1 - iaset    6 - dump     B - rx_stp | P - promis    M - coll      T - rx_err
2 - conf     7 - diag     C - retx   | O - chain     N - clr_cnt   Z - exit
3 - mcast    8 - rx_en    D - abrt   | I - init      X - exam
4 - txmit    9 - nop      E - rst    | R - repeat    K - disp
--------------------------------------------------------------------------------
>Q
 00-30   01-00
```

Figure 24

```
TOTAL TX  BAD  TX   COLLIS.   LOST CRS  UNDERUN   DEFERS    LOST CTS
000003B1  00000000  00000000  00000000  00000000  00000000  00000000
TOTAL RX  BAD  RX   CRC ERRS  OVERUN    SRT FRMS  ALG ERRS  NO EOF
000003B1  000003AF  00000000  00000000  000003AF  00000000  00000000
--------------------------------------------------------------------------------
LOOPBACK=L   CHAINING=N   PROMISCUOUS=P   REPEAT= N
--------------------------------------------------------------------------------
EXEC MACHINE       =
INTERR. PROCESSING=   TXMIT   DONE      NEW RX FRAME
STATUS  REGISTERS =   B4 00 20 24       C8 38 00 26
--------------------------------------------------------------------------------
MENU:       588 commands               |       GENERAL commands
0 - nop      5 - tdr      A - rx_dis | L - lpbck     S - stat      Q - alter
1 - iaset    6 - dump     B - rx_stp | P - promis    M - coll      T - rx_err
2 - conf     7 - diag     C - retx   | O - chain     N - clr_cnt   Z - exit
3 - mcast    8 - rx_en    D - abrt   | I - init      X - exam
4 - txmit    9 - nop      E - rst    | R - repeat    K - disp
--------------------------------------------------------------------------------
>
```

Figure 25

APPENDIX A
TROUBLESHOOTING IN CASE OF PROBLEMS

If your PC is not booting at all, you may have a mistake in the bus interface circuitry. Check the 74LS245 component and the PAL for proper enable and direction signals.

If after booting of the "E588V1" software your EXERCISER doesn't report on a CONFIG. DONE command (see figure 16) your board may have a problem.

- Check your jumpers for correct default values.
- Check your crystal oscillator for correct frequency.
- Recheck PAL equations and programming.
- Check if all integrated circuits are as specified and plugged in the correct directions.
- Check for correct resistor values.
- Check for correct capacitor values.
- Check for cold soldering joints.

APPENDIX B
VENDORS INFORMATION

PHONE JACK:

VIRGINIA PLASTICS COMPANY
3423 AERIAL WAY DR. SW.
P.O. BOX 4577
ROANOKE, VIRGINIA 24015-0577
(703) 344-0957
PART NO. 020-000-023

CABLES

BRAND-REX CORPORATION
1600 WEST MAIN ST.
WILLIMANTIC, CT 06226
(203) 423-7783
CONTACT PERSON: DEBBIE CONNERY
PART NO. BMD-04P24-J-25-DE

HUB

RETIX
1547 NINTH STREET
SANTA MONICA, CA 90401
(213) 829-4922
PART NO. HB-6, HB-12

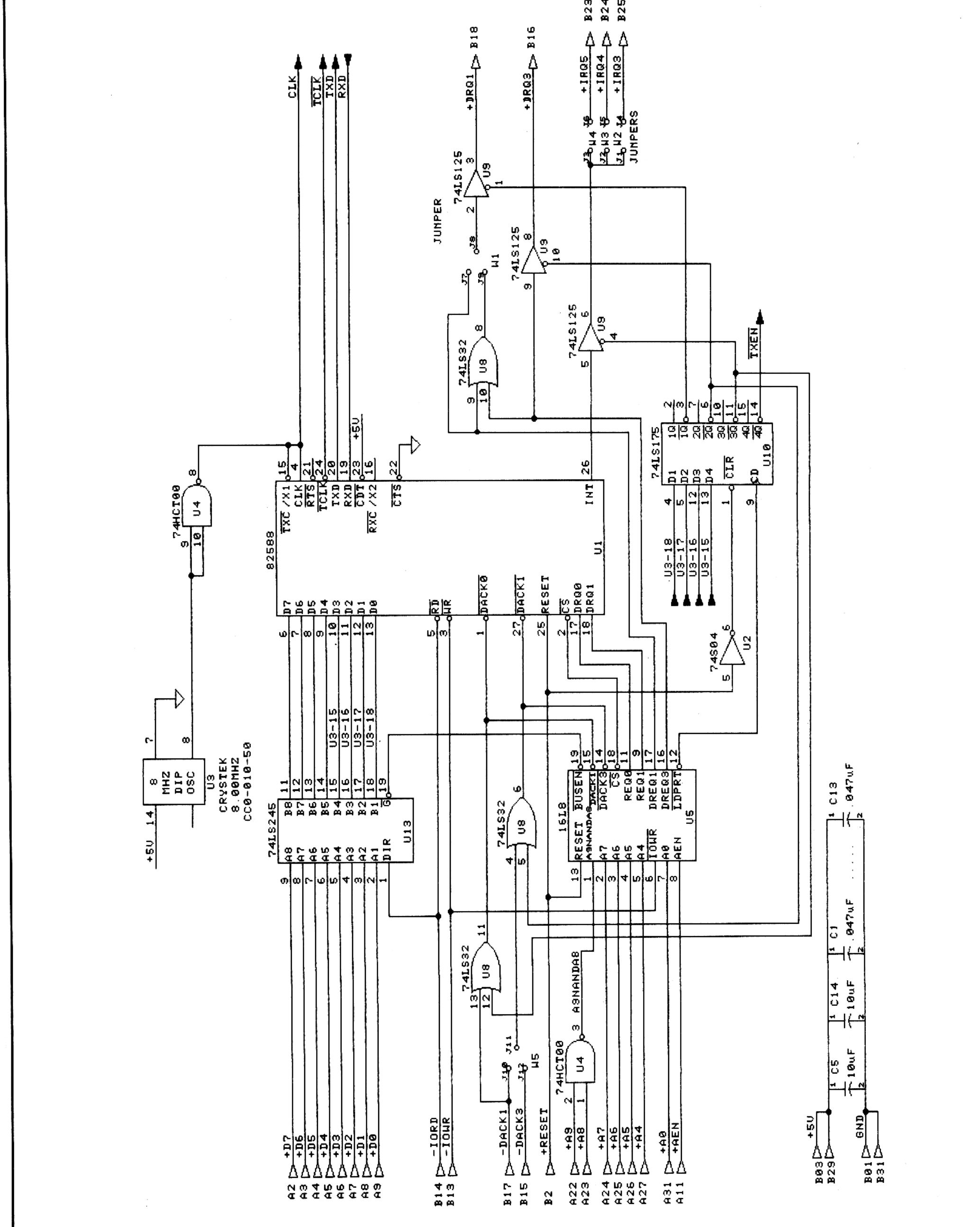
PULSE TRANSFORMER

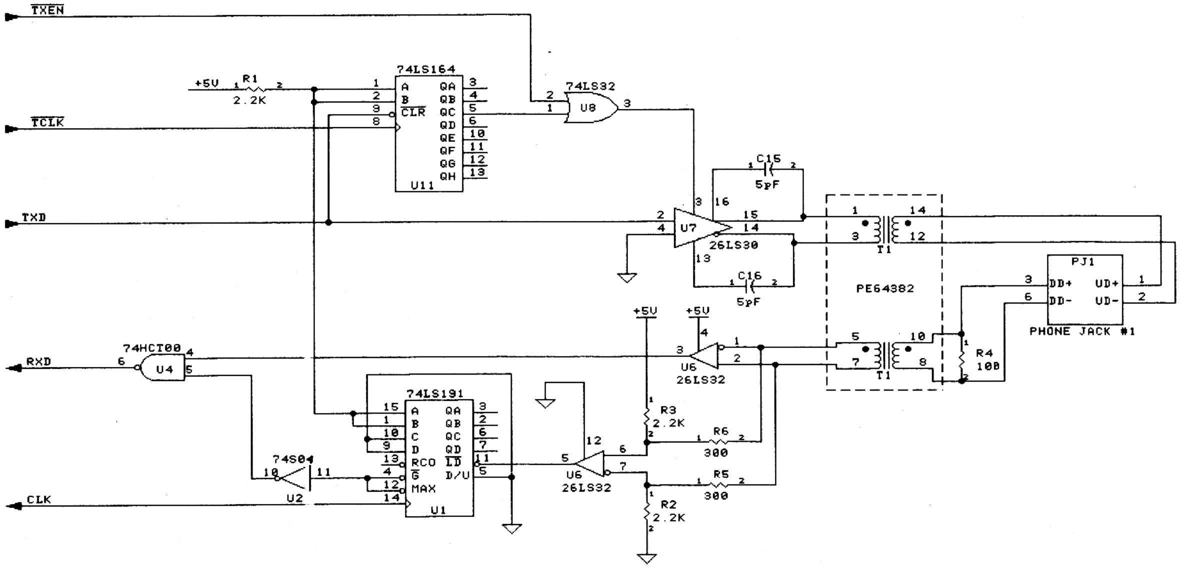
PULSE ENGINEERING
P.O. BOX 12235
SAN DIEGO, CA 92112
(619) 268-2400
PART NO. PE-64382

CRYSTAL

CRYSTAL CORPORATION
1000 CRYSTAL DRIVE
FORT MYERS, FLORIDA 33907
(813) 936-2109
PART NO. 8.000 MHZ, CCO-010-50

APPENDIX C
BOARD SCHEMATICS





EQUATIONS

$\overline{\text{CS}}$ = ! (!AEN & !A9NANDA8 & !A7 & !A6 & !A5 & !A4 & !A0) ;

$\overline{\text{LDPORT}}$ = (!AEN & !A9NANDA8 & !A7 & !A6 & !A5 & !A4 & A0 & !$\overline{\text{IOWR}}$) ;

DREQ1 = ! ((!REQ0 & !$\overline{\text{DACK1}}$) # (!REQ0 & !DREQ1) # RESET) ;

DREQ3 = ! ((!REQ1 & !$\overline{\text{DACK3}}$) # (!REQ1 & !DREQ3) # RESET) ;

$\overline{\text{BUSEN}}$ = $\overline{\text{DACK1}}$ & $\overline{\text{DACK3}}$ & (! (!AEN & !A9NANDA8 & !A7 & !A6 & !A5 & !A4)) ;

INTEL CORPORATION, 3065 Bowers Ave., Santa Clara, CA 95051; Tel. (408) 987-8080

INTEL CORPORATION (U.K.) Ltd., Swindon, United Kingdom; Tel. (0793) 488 388

INTEL JAPAN k.k., Ibaraki-ken; Tel. 029747-8511

Printed in U.S.A.